MultiSector Dynamics Community

Welcome to the newsletter of the **MultiSector Dynamics Community**

Hello MultiSector Dynamics (MSD) Community!

In this issue, we reflect on the MSD community's 2022 AGU Fall meeting presence and showcase some exciting events being planned by our working groups this Spring. We also highlight the work of Cypress Frankenfeld, a senior software engineer working with the MIT Joint Program on the Science and Policy of Global Change.

Below you will also find some general information about recent publications and job postings.

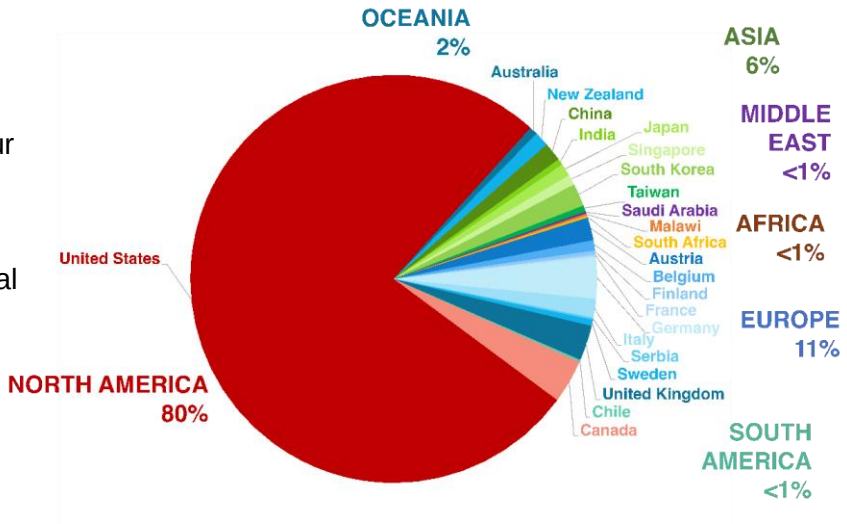
www.multisectordynamics.org

Inside This Issue

- Reflections on AGU 2022
- Research Highlight: Cypress Frankenfeld
- Upcoming WG Events
- MSD-LIVE Update
- Urban WG Webinar
- MSD Careers
- MSD Publications

Reflections on AGU 2022

Once again, MSD science had a major presence at the Fall 2022 AGU meeting, with the Community of Practice (CoP) hosting an MSD Union Session, seven oral sessions, four online poster discussions and six poster sessions. The MSD CoP sessions attracted 126 abstracts from a total of 607 authors, making it one of the largest coordinated topical areas overall in the AGU Global Environmental Change section. Over 20% of the authors of accepted oral and poster presentations came from countries other than the United States—with a total of 23 countries and all continents but Antarctica represented. Consistent with the MSD's



commitment to advance early career researchers, 34% of presenters were students. The 2022 AGU Fall Meeting MSD CoP program highlighted the value of research focused on exploring the complex sectoral interactions and potential co-evolutionary pathways within the integrated human-Earth system, including natural, engineered, and socio-economic systems. Across the sessions, several key themes and questions emerged that are relevant to advancing MSD research objectives including synergies across climate security, energy transitions, and societal aspirations, accounting for complex feedbacks, uncertainties, and thresholds when developing adaptive action pathways, or increasing resilience and advancing human well-being and environmental sustainability in urban systems.

Following the success of the MSD CoP program at the 2020, 2021 and 2022 AGU Fall Meetings, we will again coordinate several MSD session proposals for the 2023 AGU Fall Meeting, aimed at bringing together researchers from around the world, present compelling MSD research and accelerate the development of the MSD community. If you wish to contribute to this effort, please follow the **instructions to submit a 2023 AGU Fall Meeting MSD session proposals.**

MSD Research Spotlight: **Cypress Frankenfeld**

Cypress is a senior software engineer in the MIT Joint Program on the Science and Policy of Global Change. He helps build communication tools to communicate risks of climate change to stakeholders and policymakers.

I am a software engineer working with the MIT Joint Program, helping them build tools to communicate risks of climate change to state and local policymakers. One of these tools is a **website that shows an interactive county-level map of the USA**. It allows stakeholders to understand how multiple risks interact—for example, combining the percentage of people living below the poverty line with water quality reveals where people are particularly at risk of drinking low quality water, and less likely to be able to afford treatments or alternatives.

But I'm not here to write about my work making this map tool. I'm writing to the MSD newsletter to convince you that research groups can benefit from using *a software engineering mindset*. Software engineering has always been my specialty—I'm not a research scientist. There are many things that come naturally to me that may not be common-knowledge in the broader research community.

The main takeaway I want you to get from this article is that researchers could benefit by reducing *code rot*. What's code rot? Have you ever had to pick up the code of another researcher and spent hours trying to get it to run unsuccessfully? Have you ever tried to replicate a result from a simulation someone else made with the same input parameters, but it ended up with an unexpected output? This is code rot: code that doesn't run after being passed between people; code that doesn't work after being left alone for a while; code that produces different results with the same inputs when run on different computers.

Reducing code rot could help researchers in many ways, including:

1. Easy replication or extension of previous studies
2. Much quicker onboarding of new researchers
3. Less time spent fixing bugs when revisiting old code
4. Easily working with larger groups of researchers on the same model

Software engineers have had decades to come up with preventative measures for code rot, and I want to share some of the tactics that I've found most useful.

Step 1: Use version control

Version control systems—**like git**—are a practical necessity at any company with a software engineering team. They're so common that we don't even include knowledge of them in interviews—it's expected that software engineers know how to use them. Many engineers would scoff at a software company that doesn't use version control systems. In the research community there's some adoption of version control, but you can also find people writing code that's stored on someone's laptop and shared via email. Why is there such a discrepancy? I imagine part of it is the up-front investment in learning how to use git. I can say that I wholeheartedly think it's worth it for three reasons:

- You can easily roll back to working code if it breaks down the line. When I was working on the website for viewing climate change data, I tried making the map faster at one point and broke the whole thing. I felt safe to try out big changes, however, because I made them in a new branch using git, and could easily switch back to my working code with a single command.

- It helps teammates collaborate. When I was working with an undergraduate researcher, we needed to clean up some data. With the help of git and GitHub, she was able to write all the scripts for cleaning the data, and I was able to verify they worked, and provide test visualization scripts for her to test out her data-sanitizing scripts without getting out of sync.

- You can more easily share your code with the outside world. When I created a **website** to show off the results of running our EPPA model, I was able to quickly adapt a platform by the Joint Global Change Research Institute that they had added to version control and **published online**.

Step 2: Freeze the versions of libraries your code uses

The results you produce rely on a combination of the code you write and that in any third party libraries you use. The only way to guarantee repeatability is by running the exact same code, including using the same version of the same libraries. Some tools make it possible to declare exactly what library versions you use, making it easier for future researchers to run the exact same code you did. You declare these versions in a text file called a *lock file*. The lock file specifies the exact version numbers of your dependencies to be installed on any future runs. Some dependency managers that include lock files include **yarn** for JavaScript, **poetry** for python, **cargo** for Rust, **gradle** for Java, and more recently **renv** for R. I'm sure there's more examples out there but those are some good places to start.

Step 3: Always keep the latest working code in the same place

Now that you have a record of code revisions, via version control, and a way to ensure everyone is using the same versions of all the dependencies, you have a much better chance of your code continuing to work if it's working already! The last step is to make sure you always have a working version of your code that always is in the same place, so anyone visiting the code for the first time knows where to go to get off to a running start.

Git version control provides you a mechanism to save different versions of your code in separate *branches.* You can use git to create one branch that is always working, always *runnable*, let's call this branch *main.* Humans are fallible though, and can and will accidentally break the main branch. This is why software engineers often enforce this rule through automated testing and git hooks.

There are many methods I've seen software engineers use to enforce that the main branch is always runnable, but I'd like to explain my favorite at the moment. If you use GitHub, you can set up a **GitHub Action** to run and test your code when someone creates a pull request, and alter the settings of your repository to **require status checks before merging** to main. This will at least prevent people from merging unrunnable code.

Final thoughts

Using version control, freezing your dependencies with lock files, and only allowing working code in the main branch carries an upfront cost, but the great thing about it is once you've done it once, you will be able to do it again quickly by using the same tools you used for your previous project. The upfront time investment pays for itself when it saves you, or anyone else who tries to run your code years from now from the frustrations of code rot.

Avoiding code rot is particularly valuable in the field of MSD, where a main focus is on connections between different systems and sectors, often explored by linking different models. The above three suggestions all reduce the friction of making code interoperable. More interoperable and accessible code can lead to a virtuous cycle for the MSD community by making it easier for additional collaborators to join, which could open up even more opportunities to collaborate.

While I've focused solely on avoiding cod rot, there are many other aspects of the software engineering mindset that could benefit the MSD community. For example, software engineers have strategies for writing APIs that can easily talk to each other via open standards, know how to create web services that can scale when demand grows for a specific dataset or model, and are good at organizing code to be readable and simple even as the complexity of the problem it's solving grows. As the field of MSD grows, stronger connections to software engineers can help them more efficiently and effectively achieve their goals.

Key Upcoming Working Group Events

The MSD CoP is currently coordinating and planning a variety of 2023 calendar of community events for this upcoming spring.

Energy Transitions Working Group Panel: *Identifying high-priority opportunities for research in Multi-sector Impacts of Energy Transitions (MIET*)

Technological advancement, increasing demands, global climatic changes, and policy have driven rapid changes in the energy sector such as renewable energy deployment and electrification. Since the energy sector is closely interlinked with other economic sectors and natural systems, there is an increasing need to understand the resulting multisectoral impacts of energy sector transitions, along with multidirectional interactions across sectors and systems. For example, renewable fuels and electricity generation can create new competition for land and water that affects food production and urban resource use. Growth in solar and battery deployment could necessitate accelerated demand for critical materials and elements that are in limited supply or may have costly production or disposal challenges. Electrification and automation could have wide-ranging impacts on fuel use, mobility, and energy consumption throughout the world, with additional interactions with water and land use. This webinar will discuss current understanding of multisectoral relationships and identifies gaps in our current capability to model and analyze how energy sector transitions interact with the rest of society and nature. The panelists will share their insights about the interactions, sectors, and systems that are missing or inadequately addressed by existing analytical approaches, and propose new research pathways and community approaches that could help build a more holistic understanding of energy transitions. **The panel will be held on March 10th from 3-4:15 PM ET over Zoom. Links to register coming soon!**

The Working Group on Uncertainty Quantification and Scenario Development Seminar: *Dr. Afreen Siddiqi*

On **March 17th at 1:00 PM ET,** the UQ/SD working group will host a webinar presentation by Dr. Afreen Siddiqi on approaches for exploring uncertainty in waste-to-energy and water infrastructure planning problems using complex adaptive systems theories. This will be followed in May by a community discussion of key emerging themes and research needs in MSD to establish topics and priorities for upcoming working group events and workshops. Details on both events will be provided through the

UC/UQ mailing list and website. If you are not already subscribed to our mailing list, you can join here **https://multisectordynamics.org/join-us/**. **Links to register for the webinar coming soon!**

Recent Urban Working Group Hosted Seminar

Dr. Yuyu Zhou is an Associate Professor in the Department of Geological and Atmospheric Sciences at Iowa State University. His research interests lie in the applications of geospatial technologies and integrated assessment modeling to understanding the problems of global environmental change and their potential solutions. On January 15th, he presented a seminar titled **Urban Environmental Change and its Implications for Building Energy Demand** to the Urban Working Group. In this seminar, Dr. Zhou described his recent work characterizing historical and future urban dynamics as well as the associated urban environmental changes from local to global scales using satellite remote sensing observations and developed modeling capabilities. He also presented research estimating the impact of urban environmental change on building energy demand. The products, models, and findings from these studies are of great value for advancing our understanding of the coupled urban-nature system under urbanization for sustainability in the context of climate change. Twenty-five people from the Multi-Sector Dynamics community, DOE National Laboratories, and universities attended the seminar, which concluded with a robust series of questions and discussion with the audience.

Update on MSD-LIVE

MSD-LIVE launched to the community in August of last year. Since then, we've seen great engagement with the platform including more than 60 registered users and 28 datasets either drafted or published in final form. Please reach out to the MSD-LIVE team by emailing info@msdlive.org if you have questions about how to use MSD- LIVE to manage your data.

7	Projects
64	Registered Users
11	Published Datasets
17	Draft Datasets
>185 Tb	Data Volume

MSD Job Listings

Our website features a **careers page** that lists available MSD-focused positions at all levels. If you'd like to post a position to be featured in this page, please email us at: contact@multisectordynamics.org. Here are some of our latest postings:

Associate Director, Office of Biological and Environmental Research

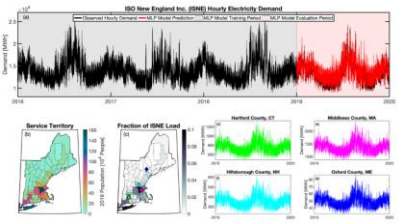
The Department of Energy's (DOE) Office of Science - Headquarters is looking for a dynamic, innovative, seasoned executive to lead the Office of Biological and Environmental Research (BER). The Office of Science is the largest supporter of basic research in the physical sciences in the United States, providing more than 40 percent of total funding in this area.

Physical Scientist Position (Office of Biological and Environmental Research; U.S. Department of Energy)

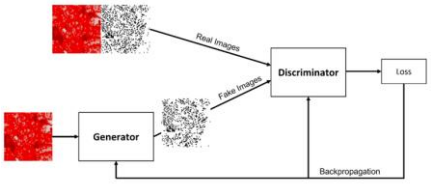
The Department of Energy is seeking motivated and highly-qualified candidates for exciting positions available in multiple locations throughout the United States. This series includes positions that involve professional work in the physical sciences and may include work in a combination of physical science fields.

Highlighted Publications

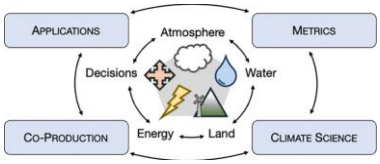
We have been posting and will be regularly updating select MSD relevant publications on the website, under the Publications page. If you have any publications you would like us to highlight, please email contact@multisectordynamics.org.
Below you can find some of the publications posted most recently:



tell: a Python package to model future total electricity loads in the United States



Generative Adversarial Networks for Ensemble Projections of Future Urban Morphology



Metrics as tools for bridging climate science and applications

---

This newsletter has been edited by Rohini Gupta and the Community of Practice Facilitation Team. This and all previous newsletters can be accessed at the Newsletters page of our website. If you have any suggestions, concerns or other feedback about this newsletter or the MSD website, please email contact@multisectordynamics.org.